Amendments to the Claims:

This listing of claims replaces all prior listings of claims:

Listing of Claims:

1. (Previously Presented) A computer-implemented method comprising:

providing a design-time translator and a run-time translator that both correspond to a defined page element, the run-time translator and design time translator configured on a processor;

during design-time for a page, invoking the design-time translator for a page template including the defined page element having content components, said design-time invoking resulting in the defined page element in the page template being translated into a design-time representation of the content components in the page, the design-time representation being rendered in accordance with a predefined layout of a container for the content components, the page template being available to a plurality of remote users of a portal, the content components including a first content component and a second content component, the first content component configured as static content with a run-time behavior determinable at design-time, and the second component configured as dynamic content with a run-time behavior not determinable at run-time, such that at design-time a tag is used to represent the dynamic content on the page rendered at design-time; and

during run-time for the page, invoking the run-time translator for the page template, said run-time invoking resulting in the content components being obtained and the defined page element in the page template being translated into a run-time presentation of the obtained one or more content components in accordance with the layout of the container, wherein the second component configured as dynamic content is determined and obtained in

parallel, at run-time using threading, with other dynamic content stored in blocks without ordering in a content storage medium to render the dynamic content of the second component rather than the tag used during design-time.

2. (Previously Presented) The method of claim 1, wherein said invoking the design-time translator further results in presentation of a WYSIWYG layout editor using the design-time representation of the one or more content components in the page.

3. (Previously Presented) The method of claim 2, wherein the said invoking the design-time translator further results in client-side scripting components being included in the design-time representation to form at least part of the WYSIWYG layout editor and enable adding a content component to a content container using a drag-and-drop action.

4. (Original) The method of claim 2, wherein the page template comprises a portal page template, and the WYSIWYG layout editor comprises a WYSIWYG portal page layout editor.

5. (Original) The method of claim 4, wherein the defined page element comprises a custom Java Server Page tag and the design-time translator and the run-time translator comprise Java Server Page tag handlers for the custom Java Server Page tag, and wherein the run-time translator obtains portal dynamic content according to the portal page template and the design-time translator does not.

6. (Previously Presented) An article comprising a machine-readable storage medium storing instructions operable to cause one or more machines to perform operations comprising: providing a design-time translator and a run-time translator that both correspond to a

defined page element, the run-time translator and design time translator configured on a processor;

during design-time for a page, invoking the design-time translator for a page template including the defined page element having-content components, said design-time invoking resulting in the defined page element in the page template being translated into a design-time representation of the content components in the page, the design-time representation being rendered in accordance with a predefined layout of a container for the content components, the page template being available to a plurality of remote users of a portal, the content components including a first content component and a second content component, the first content component configured as static content with a run-time behavior determinable at design-time, and the second component configured as dynamic content with a run-time behavior not determinable at run-time, such that at design-time a tag is used to represent the dynamic content on the page rendered at design-time; and

during run-time for the page, invoking the run-time translator for the page template, said run-time invoking resulting in the content components being obtained and the defined page element in the page template being translated into a run-time presentation of the obtained one or more content components in accordance with the layout of the container, wherein the second component configured as dynamic content is determined and obtained in parallel, at run-time using threading, with other dynamic content stored in blocks without ordering in a content storage medium to render the dynamic content of the second component rather than the tag used during design-time.

7. (Original) The article of claim 6, wherein translating the placeholder during design-time comprises adding code enabling editing of the portal page, the added code forming at least part of the WYSIWYG portal layout editor.

8. (Original) The article of claim 7, wherein the added code comprises client-side scripting that enables addition of a content component to a content container in the portal page using a drag-and-drop action.

9. (Original) The article of claim 6, wherein the placeholder comprises a custom Java Server Page tag, said translating the placeholder during design-time comprises invoking a design-time Java Server Page tag handler corresponding to the custom Java Server Page tag, and said translating the placeholder during run-time comprises invoking a run-time Java Server Page tag handler corresponding to the custom Java Server Page tag.

10-17 (Canceled).

18. (Previously Presented) A portal system comprising:

a processor; and

memory configured to:

provide a design-time translator and a run-time translator that both correspond to a defined page element, the run-time translator and design time translator configured on a processor;

during design-time for a page, invoke the design-time translator for a page template including the defined page element having content components, said design-time invoking resulting in the defined page element in the page template being translated into a design-time

representation of the content components in the page, the design-time representation being rendered in accordance with a predefined layout of a container for the content components, the page template being available to a plurality of remote users of a portal, the content components including a first content component and a second content component, the first content component configured as static content with a run-time behavior determinable at design-time, and the second component configured as dynamic content with a run-time behavior not determinable at run-time, such that at design-time a tag is used to represent the dynamic content on the page rendered at design-time; and

during run-time for the page, invoke the run-time translator for the page template, said run-time invoking resulting in the content components being obtained and the defined page element in the page template being translated into a run-time presentation of the obtained one or more content components in accordance with the layout of the container, wherein the second component configured as dynamic content is determined and obtained in parallel, at run-time using threading, with other dynamic content stored in blocks without ordering in a content storage medium to render the dynamic content of the second component rather than the tag used during design-time.

19. (Original) The system of claim 18, wherein the first tag handler interprets the portal page template by including client-side scripting that enables addition of a content component to a content container in the portal page template using a drag-and-drop action.

20. (Original) The system of claim 18, wherein the defined page element comprises a custom Java Server Page tag and the design-time translator and the run-time translator comprise Java Server Page tag handlers for the custom Java Server Page tag, and

wherein the run-time translator obtains portal dynamic content according to the portal page template and the design-time translator does not.

21-25. (Canceled).

26. (Currently Amended) A computer-implemented method for selectively interpreting a portal page layout template, the method comprising:

providing a design-time translator and a run-time translator, the design-time translator and the run-time translator both corresponding to a same defined page element or placeholder, and being invoked based on a current mode of operation, the run-time translator and design time translator configured on a processor;

translating a placeholder in a portal template during design-time into a representation of a container designed to present portal content using a single template file for both run-time and design-time, the container representation showing a layout context for the portal content that will be obtained and revealed at run-time, the container representation also directly presenting dynamic content source information for the content container;

presenting a WYSIWYG portal layout editor using the container representation designed to present the portal dynamic content, the WYSIWYG portal layout editor facilitating editing of the portal template and the resulting portal page;

obtaining portal dynamic content during run-time from a dynamic content source, the placeholder in the portal template being translated by the runtime translator into a presentation of the container containing the obtained portal dynamic content component, wherein the dynamic content is determined and obtained in parallel, at run-time using threading, with other dynamic content stored in blocks without ordering in a dynamic content source, providing

a storage medium, to render the dynamic content rather than a tag used during design-time to represent the dynamic content; and

parsing and locating, by a run-time application, the placeholders in the template and replacing them with the run-time content components, and at design-time, parsing and rendering the same template with a representation of the content components, in place of the actual run-time content components, to reveal the run-time layout during design of the template.